

REMARKS

Claims 1, 4-8, 11-15, 18-19, 22-28, 30-32, 34-36 and 38-39 are pending in the present application. Applicants respectfully request reconsideration of the application in view of the amendments and remarks made herein.

I. Rejections Under 35 U.S.C. § 103

Claims 1, 4-8, 11-15, 18-19, 22-28, 30-32, 34-36 and 38-39 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Cavanaugh* (US 5,809,507) in view of *Bannon* (US 5,297,279). The Examiner essentially stated that the combined teachings of *Cavanaugh* and *Bannon* teach or suggest all of the limitations of Claims 1, 4-8, 11-15, 18-19, 22-28, 30-32, 34-36 and 38-39.

Claims 1, 15, 26 and 38 are the independent claims.

Claims 1 and 38 claim, and Claims 15 and 26 essentially claim, *inter alia*, “generating automatically, by the processor, a database table within the persistent storage structure to store the object instance data, wherein the database table comprises a column for each property of the object and the column includes a counter that is incremented each time the corresponding property is accessed.”

Cavanaugh teaches writing object attributes to a persistent storage mechanism (i.e., a data store) (see col. 8, lines 54-60). *Cavanaugh* does not teach or suggest “generating automatically, by the processor, a database table within the persistent storage structure to store the object instance data, wherein the database table comprises a column for each property of the object and

the column includes a counter that is incremented each time the corresponding property is accessed” (emphasis added) as claimed in Claims 1 and 38 and essentially as claimed in Claims 15 and 26. In the Advisory Action, the Examiner cites col. 8, lines 55-63 of *Cavanaugh* as teaching storing information in a data store on a persistent storage medium; the Examiner asserts that this is analogous to generating a database table within a persistent storage structure, essentially as claimed in Claims 1, 15, 26 and 38. Respectfully, in *Cavanaugh*, the data structure of the data store used in the persistent storage mechanism is a hierarchical tree. Each object (504) is stored in a node of the tree, and the properties (506) of each object are stored in corresponding sub-nodes of the tree (see col. 12, lines 24-44 and FIG. 5). By way of comparison, Claims 1, 15, 26 and 38 claim storing objects in a database table, and storing the properties of the object in respective columns. Further, nowhere does *Cavanaugh* teach or suggest including a counter with each property in the data structure, nor does *Cavanaugh* teach or suggest incrementing the counter (e.g., measuring the frequency of access of the properties of an object), essentially as claimed in Claims 1, 15, 26 and 38. Indeed, because *Cavanaugh* already indexes every object within a data store, there is no need for *Cavanaugh* to create indices for certain properties of the objects based on a predefined threshold. Thus, *Cavanaugh* fails to teach or suggest all of the limitations of Claims 1, 15, 26 and 38.

Bannon teaches storing persistent objects in a persistent object server (POS server) (54) located within an object-oriented database (OODB) (18) (see FIG. 2; col. 10, lines 11-15; col. 8, lines 20-23). *Bannon* does not teach or suggest “generating automatically, by the processor, a database table within the persistent storage structure to store the object instance data, wherein the database table comprises a column for each property of the object and the column includes a counter that is incremented each time the corresponding property is accessed” as claimed in

Claims 1 and 38 and essentially as claimed in Claims 15 and 26. Indeed, *Bannon* is silent on the data structure of the POS server, and does not discuss utilizing a database table with a column – including a counter – for each property of an object. Thus, *Bannon* fails to cure the deficiencies of *Cavanaugh*.

The combination of *Cavanaugh* and *Bannon* teaches writing object attributes to a persistent object server (POS server) that utilizes a hierarchical tree as a data structure. The combination does not teach or suggest “generating automatically, by the processor, a database table within the persistent storage structure to store the object instance data, wherein the database table comprises a column for each property of the object and the column includes a counter that is incremented each time the corresponding property is accessed” as claimed in Claims 1 and 38 and essentially as claimed in Claims 15 and 26. Accordingly, the combination does not teach or suggest all of the limitations of Claims 1, 15, 26 and 38.

Claims 1 and 38 further claim, and Claims 15 and 26 further essentially claim, *inter alia*, “generating automatically, by the processor, an index to a property of the object if the counter corresponding to the property exceeds a predefined threshold.”

Cavanaugh teaches an index to all objects in a data store and a use count to determine whether the index is invalid (see col. 17, lines 34-54). *Cavanaugh* does not teach or suggest “generating automatically, by the processor, an index to a property of the object if the counter corresponding to the property exceeds a predefined threshold” as claimed in Claims 1 and 38 and essentially as claimed in Claims 15 and 26. Consider that in *Cavanaugh*, an index (1016) to each object in a data store is created, said creation having no connection to a counter, wherein a use

count (1018) is incremented each time the corresponding index (1016) is reused, in order to determine whether the index (1016) is still valid (see col. 17, lines 34-54 and FIG. 10). By way of comparison, the index in Claims 1, 15, 26 and 38 is generated for a property of an object, and is only generated when a predefined threshold is exceeded. Further, while the use count in *Cavanaugh* is utilized after the index has already been created in order test the validity of the existing index, the counter in Claims 1, 15, 26 and 38 is utilized before the index is created in order to determine whether the index should be created. Thus, *Cavanaugh* clearly does not teach or suggest all of the limitations of Claims 1, 15, 26 and 38.

Bannon teaches an object-oriented database (OODB) for providing long-term storage and retrieval of objects created by application programs written at least in part in object-oriented programming languages (see col. 5, lines 39-45). *Bannon* does not teach or suggest “generating automatically, by the processor, an index to a property of the object if the counter corresponding to the property exceeds a predefined threshold” as claimed in Claims 1 and 38 and essentially as claimed in Claims 15 and 26. Indeed, *Bannon* is silent on utilizing an index with a counter in any manner. Thus, *Bannon* fails to cure the deficiencies of *Cavanaugh*.

The combination of *Cavanaugh* and *Bannon* teaches an object-oriented database (OODB), an index to all objects in the OODB, and a use count to determine whether the index is invalid. The combination does not teach or suggest “generating automatically, by the processor, an index to a property of the object if the counter corresponding to the property exceeds a predefined threshold” as claimed in Claims 1 and 38 and essentially as claimed in Claims 15 and 26. Accordingly, the combination does not teach or suggest all of the limitations of Claims 1, 15, 26 and 38.

Therefore, for at least the reasons above, Claims 1, 15, 26 and 38 are believed to be patentable and non-obvious over the combination of *Cavanaugh* and *Bannon*. Applicants respectfully submit that inasmuch as Claims 4-8, 11-14, 18-19, 22-25, 27-28, 30-32, 34-36 and 39 are dependent on Claims 1, 15, 26 and 38, and Claims 1, 15, 26 and 38 are patentable over the cited references, Claims 4-8, 11-14, 18-19, 22-25, 27-28, 30-32, 34-36 and 39 are allowable as being dependent on allowable independent claims. Withdrawal of the instant rejection is respectfully requested.

CONCLUSION

In view of the foregoing, it is believed that all claims now pending patentability define the subject invention over the prior art of record and are in condition for allowance.

Early and favorable reconsideration of the case is respectfully requested.

Respectfully submitted,

Date: December 28, 2009

By: /Nathaniel T. Wallace/
Nathaniel T. Wallace
Reg. No. 48,909
Attorney for Applicant(s)

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889